	KEMENTRIAN PENDIDIKAN DAN KEBUDAYAAN UNIVERSITAS BRAWIJAYA FAKULTAS TEKNIK JURUSAN TEKNIK ELEKTRO Jalan MT Haryono 167 Telp & Fax. 0341 554166 Malang 65145	**KODE** **PJ-01**

PENGESAHAN
PUBLIKASI HASIL PENELITIAN SKRIPSI
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS BRAWIJAYA

NAMA : NILUH KADEK KURNIA DEWI
NIM : 0610630072 - 63
PROGRAM STUDI : REKAYASA KOMPUTER
JUDUL SKRIPSI : IMPLEMENTASI ALGORITMA RC6 UNTUK PROTEKSI *FILE*MP3

TELAH DI-REVIEW DAN DISETUJUI ISINYA OLEH:

Pembimbing 1	Pembimbing 2
Ir. Muhammad Aswin	**Waru Djuriatno, ST., MT.**
NIP. 19640626 199002 1 001	**NIP. 19690725 199702 1 001**

IMPLEMENTASI ALGORITMA RC6 UNTUK PROTEKSI *FILE* MP3

Publikasi Jurnal Skripsi

Disusun Oleh :

NILUH KADEK KURNIA DEWI

NIM : 0610630072 - 63

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2013

IMPLEMENTASI ALGORITMA RC6 UNTUK PROTEKSI *FILE* MP3

Niluh Kadek K. D.[1], Ir. Muhammad Aswin [2], Waru Djuriatno, ST., MT.[2]
[1]Mahasiswa Teknik Elektro Univ. Brawijaya, [2]Dosen Teknik ElektroUniv. Brawijaya
Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya
Jalan MT. Haryono 167, Malang 65145, Indonesia
E-mail: niluhkurniadewi@gmail.com

***Abstract*—Nowadays, audio digital file such as MP3 is popular and easy to play. MP3 file both easy to distribute and easy to duplicate without regard the copyright. Therefore, we need an application to protect MP3 file such as encryption. The encrypted MP3 file can not played without this player which can decrypt it. This application use RC6 algorithm to encrypt and decrypt. RC6 algorithm is the simple algorithm which use private key and iterated cipher for its security.**
***Index Terms*— MP3, Encryption, Decryption.**

***Abstrak*—Penggunaan *file* berupa *audio digital* berformat MP3 saat ini cukup populer dan mudah untuk dinikmati. *File* MP3 selain memberi kemudahan dalam penyebaran, juga memberi kemudahan dalam penggandaan yang kemudian dapat digunakan secara negatif tanpa memperhatikan aspek hak cipta. Untuk itu diperlukan aplikasi untuk memproteksi *file* MP3 yaitu dengan enkripsi. *File* MP3 yang telah terenkripsi tidak dapat diputar/dimainkan dengan sempurna sehingga diperlukan aplikasi untuk mendekripsi *file* tersebut agar dapat dimainkan seperti semula. Aplikasi dirancang dengan menggunakan algoritma RC6 untuk melakukan enkripsi dan dekripsi agar keamanan dapat ditingkatkan. Algoritma RC6 adalah suatu algoritma kunci privat yang dikenal dengan kesederhanaannya. Algoritma RC6 merupakan algoritma dengan parameter yang dapat bekerja pada panjang kunci yang beragam. Untuk aspek keamanannya, algoritma RC6 mengutamakan prinsip iterated cipher.**
***Kata Kunci*—MP3, Enkripsi, Dekripsi.**

I. PENDAHULUAN

Multimedia dapat diartikan sebagai teknologi yang menggabungkan berbagai sumber media (teks, grafik dan suara) untuk menyampaikan atau membuat sesuatu sebagai perantara atau suatu bentuk komunikasi. Multimedia seringkali digunakan dalam dunia hiburan. Salah satunya adalah penggunaan *file* berupa *audio digital* yang saat ini cukup populer dan mudah untuk dinikmati, yaitu *file* berformat MP3.

File MP3 selain memberi kemudahan dalam penyebaran, juga memberi kemudahan dalam penggandaan. Kemudahan tersebut akhirnya dapat digunakan secara negatif tanpa memperhatikan aspek hak cipta. *File* MP3 yang seharusnya menjadi properti legal dari produsen dan secara legal dimiliki oleh orang yang telah membelinya bisa dengan mudah disalahgunakan oleh pihak-pihak yang tidak bertanggung jawab. Untuk itu diperlukan suatu cara untuk memproteksi *file* tersebut, salah satunya adalah dengan enkripsi. *File* MP3 yang telah terenkripsi tidak dapat diputar/dimainkan dengansempurna sehingga diperlukan aplikasi untuk mendekripsi*file* tersebut agar dapat dimainkan seperti semula.

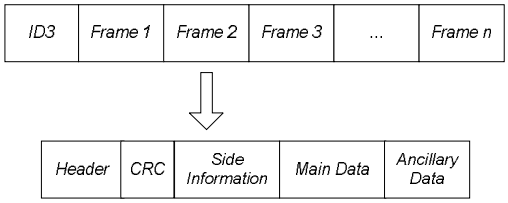
II. TINJAUAN PUSTAKA

A. MPEG-1 Layer 3 (MP3)

MPEG-1 Audio Layer 3 atau lebih dikenal sebagai MP3 adalah salah satu format berkas pengodean suara yang memiliki kompresi yang baik (meskipun bersifat *lossy*) sehingga ukuran berkas bisa memungkinkan menjadi lebih kecil. Berkas ini dikembangkan oleh seorang insinyur JermanKarlheinz Brandenburg. MP3 memakai pengodean *Pulse Code Modulation* (PCM). MP3 mengurangi jumlah *bit* yang diperlukan dengan menggunakan model *psychoacoustic* untuk menghilangkan komponen-komponen suara yang tidak terdengar oleh manusia.

File MP3 tersusun dari banyak *frame* MP3, yang terdiri dari sebuah *header* dan sebuah blok data. Setiap frame secara umum menyimpan 1152 sampel audio selama 26 ms. Artinya frame rate yang dihasilkan sekitar 38 fps. Dengan tambahan setiap frame dibagi menjadi 2 unit yang menyimpan 576 sampel. Karena *bitrate*

menentukan ukuran setiap sampel maka memperbesar *bitrate* akan memperbesar ukuran dari frame tersebut. Sebuah frame terdiri dari lima bagian yaitu *header* , CRC, side information, main data dan terakhir ancillary data [3].



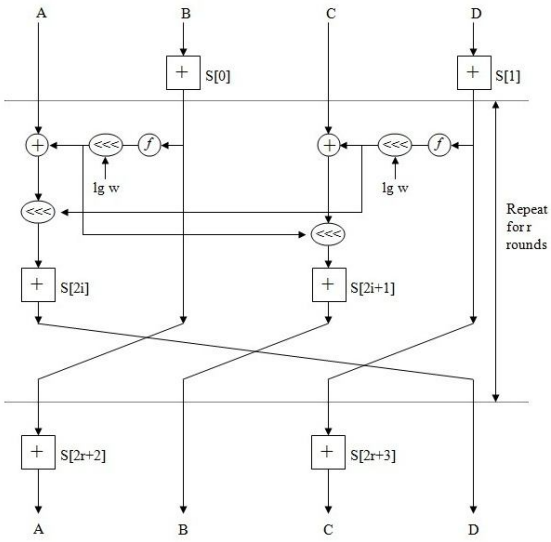
Gambar 1. Struktur *Frame* MP3

B. Algoritma RC6

RC6 merupakan salah satu dari algoritma simetri kriptografi yaitu algoritma yang menggunakan satu kunci untuk enkripsi dan dekripsinya. RC6 adalah algoritma blok kode yang sangat aman, padat, sederhana dan menawarkan performansi yang sangat bagus dan fleksibel, dikembangkan dari algoritma RC5 [2].

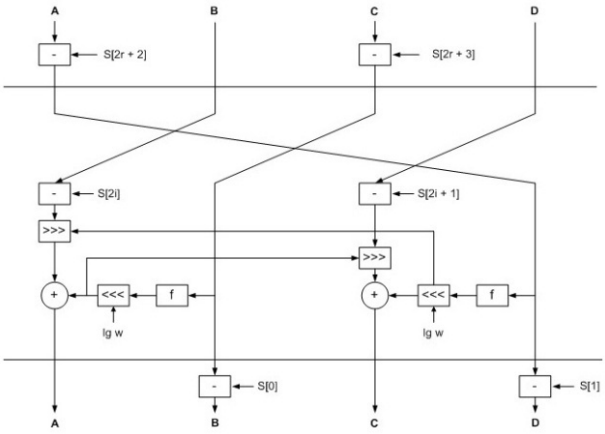
Algoritma RC6 adalah versi yang dilengkapi dengan beberapa parameter, sehingga dituliskan sebagai RC6-w/r/b, dimana parameter w merupakan ukuran kata dalam satuan bit, r adalah bilangan bulat bukan negatif yang menunjukan banyaknya iterasi selam proses enkripsi, dan b menunjukan ukuran kunci enkripsi dalam *byte*. Ketika algoritma ini masuk sebagai kandidat AES, maka ditetapkan nilai parameter w = 32, r = 20 dan b bervariasi antara 16, 24, dan 32 *byte.*

Karena RC6 memecah blok 128 bit menjadi 4 buah blok 32 bit, maka algoritma ini bekerja dengan 4 buah register 32-bit A, B, C, D. *Byte* yang pertama dari plaintext atau ciphertext ditempatkan pada *byte* A, sedangkan *byte* yang terakhirnya ditempatkan pada *byte* D. Dalam prosesnya akan didapatkan (A, B, C, D) = (B, C, D, A) yang diartikan bahwa nilai yang terletak pada sisi kanan berasal dari register disisi kiri. Diagram blok berikut akan lebih menjelaskan proses enkripsi yang terjadi pada algoritma RC6 :



Gambar 2. Proses Enkripsi Algoritma RC6

Proses dekripsi ciphertext pada algoritma RC6 merupakan pembalikan dari proses enkripsi. Pada proses whitening, bila proses enkripsi menggunakan operasi penjumlahan, maka pada proses dekripsi menggunakan operasi pengurangan. Sub kunci yang digunakan pada proses whiteneing setelah iterasi terakhir diterapkan sebelum iteasi pertama, begitu juga sebaliknya sub kunci yang diterapkan pada proses whitening sebelum iterasi pertama digunakan pada whitening setelah iterasi terkahir. Akibatnya, untuk melakukan dekripsi, hal yang harus dilakukan semata-mata hanyalah menerapkan algoritma yang sama dengan enkripsi, dengan tiap iterasi menggunakan sub kunci yang sama dengan yang digunakan pada saat enkripsi, hanya saja urutan sub kunci yang digunakan terbalik. Diagram blok berikut akan lebih menjelaskan proses dekripsi yang terjadi pada algoritma RC6 :



Gambar 3. Proses Dekripsi Algoritma RC6

Pengguna memasukkan sebuah kunci yang besarnya b *byte*, dimana 0≤b≤255. *byte* kunci ini

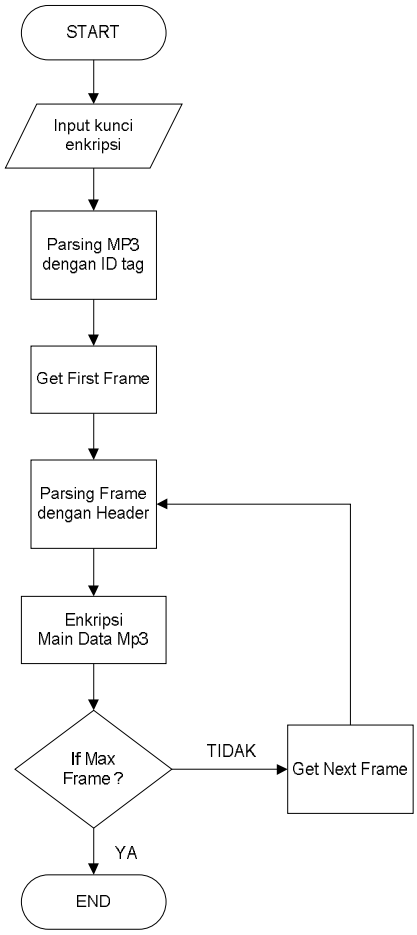
kemudian ditempatkan dalam array c w-bit words L[0]…L[c-1]. *Byte* pertama kunci akan ditempatkan sebagai pada L[0], *byte* kedua pada L[1], dan seterusnya. (Catatan, bila b=0 maka c=1 dan L[0]=0). Masing-masing nilai kata w-bit akan dibangkitkan pada penambahan kunci round 2r+4 dan akan ditempatkan pada array S[0,…,2r+3].

Konstanta P32 = B7E15163 dan Q32 = 9E3779B9 (dalam satuan heksadesimal) adalah "konstanta ajaib" yang digunakan dalam penjadwalan kunci pada RC6. nilai P32 diperoleh dari perluasan bilangan biner e-2, dimana e adalah sebuah fungsi logaritma. Sedangkan nilai Q32 diperoleh dari perluasan bilangan biner ∅-1, dimana ∅ dapat dikatakan sebagai "golden ratio" (rasio emas) [1].

III. PERANCANGAN APLIKASI

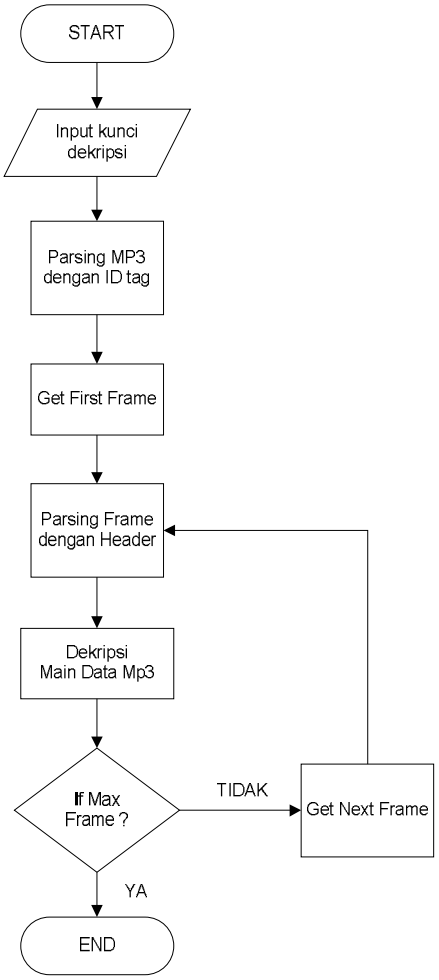
Cara kerja aplikasi ini dimulai dengan input file MP3 yang ingin diproteksi lalu *user* menentukan dan meng-*input* kunci/*password*. Proses selanjutnya terjadi dalam aplikasi, kunci/*password* yang dimasukkan oleh *user* dibangkitkan menjadi sub kunci kemudian digunakan dalam proses enkripsi yang kemudian menghasilkan keluaran berupa file MP3 terproteksi (*chiper*). Apabila *user* ingin mengembalikan file MP3 yang terproteksi tersebut seperti semula, *user* kembali memasukkan file MP3 tersebut ke dalam aplikasi kemudian meng-*input* kunci/*password* maka proses dekripsi akan terjadi dan menghasilkan keluaran berupa file MP3 asli (*plain*).

a) Proses enkripsi file MP3
   Proses enkripsi merupakan cara untuk mendapatkan *cipher* dari *plaintext* asli. Pada proses sebelumnya diasumsikan bahwa tabel sub kunci yang baru telah didapatkan sehingga tabel sub kunci yang baru tersebut akan digunakan dalam proses perhitungan untuk mengacak *plaintext* asli agar dapat diubah menjadi *cipher*. Berikut ini adalah perancangan enkripsi :



Gambar 5. Proses Enkripsi *File* MP3

b) Proses dekripsi file MP3
   Proses dekripsi merupakan cara untuk mengembalikan *cipher* ke *plaintext* asli. Pada proses ini digunakan kunci yang telah dibangkitkan seperti pada proses enkripsi. Berikut ini adalah perancangan dekripsi :



Gambar 6. Proses Dekripsi *File* MP3

IV. PENGUJIAN

A. Pengujian Validasi

Pengujian dilakukan untuk memastikan bahwa aplikasi dapat memenuhi kebutuhan fungsional untuk melakukan enkripsi *file* MP3 menjadi acak dan dekripsi *file* MP3 acak menjadi *plaintext*. Berikut ini adalah hasil pengujian yang diperoleh:

Tabel 1.*Test case* untuk pengujian validasi

No	Kasus Uji	Hasil yang didapatkan	Status
1.	Enkripsi pesan rahasia	Aplikasi dapat melakukan proses dekripsi dengan hasil data pada tabel enkripsi No | Cipher 1 | 7714C4FCFF24B9A8 CA322FF181BAB98D 2 | A152E0EFE1B0FEBC 08F133F47FFC86DB 3 | 76DE28232E6AC25E2 BEC5EA9A983C736	Valid
2	Dekripsi pesan rahasia	Aplikasi dapat melakukan proses dekripsi dan mengembalikan ciphertext ke plaintext seperti pada data enkripsi No | Hasil dekripsi 1 | 8004035BD3D30F1B6 A3B838A7965E849 2 | 8C34B566EC3D0B61 86222CDD860974D7 3 | 0B848D57A7B1A961 698F2B34F6252454	Valid

B. Pengujian Kualitas Suara

Pengujian ini bertujuan untuk membandingkan kualitas suara *file* MP3 setelah dienkripsi dan setelah didekripsi kembali, apakah dapat didengarkan dengan baik atau tidak. Pengujian ini dilakukan secara subyektif dengan menggunakan dua sampel data yang berbeda.

Berikut ini adalah hasil pengujian yang diperoleh:

Tabel 2. Hasil Pengujian Kualitas Suara File MP3 Setelah Proses Enkripsi

	Berkas 1	Berkas 2
Subyek 1	Tidak dapat	Tidak dapat
	didengarkan	didengarkan
Subyek 2	Tidak dapat	Tidak dapat
	didengarkan	didengarkan
Subyek 3	Tidak dapat	Tidak dapat
	didengarkan	didengarkan
Subyek 4	Tidak dapat	Tidak dapat
	didengarkan	didengarkan

Tabel 3. Hasil Pengujian Kualitas Suara File MP3 Terenkripsi Setelah Proses Dekripsi

	Berkas 1	Berkas 2
Subyek 1	Terdengar sama	Terdengar sama
Subyek 2	Terdengar sama	Terdengar sama
Subyek 3	Terdengar sama	Terdengar sama
Subyek 4	Terdengar sama	Terdengar sama

C. Pengujian Kecepatan Proses

Pengujian kecepatan proses pada aplikasi ini ditujukan untuk mengetahui proses enkripsi dan dekripsi sebagai acuan analisis terhadap kecepatan proses.

Tabel 4. Data hasil pengujian kecepatan proses enkripsi

panjang plain (byte)	panjang kunci (byte)	waktu (ms)	waktu (ms)	waktu (ms)	rata-rata
161385	5	776	701	720	732.3333
321463	10	1313	942	960	1071.6667
481542	15	1564	1529	1341	1478
641202	20	1782	1880	1544	1735.3333
801281	25	2287	1995	1949	2077

Tabel 5. Data hasil pengujian kecepatan proses dekripsi

chiper (byte)	panjang kunci (byte)	waktu (ms)	waktu (ms)	waktu (ms)	rata-rata
161385	5	463	507	584	518
321463	10	719	716	999	811.3333
481542	15	1361	1394	1334	1363
641202	20	1521	1552	1486	1519.6667
801281	25	1741	1781	1913	1811.6667

V. PENUTUP

A. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, dapat diambil kesimpulan bahwaterjadi penurunan kualitas suara setelah *file* MP3 dienkripsi sehingga tidak dapat didengarkan dengan baik oleh telinga manusia. Namun setelah didekripsi, *file* MP3 tersebut dapat didengarkan kembali dan memiliki kualitas suara yang hampir sama dengan *file* aslinya.

Proses dekripsi dan enkripsi menunjukan bahwa semakin panjang plaintext yang dimasukan atau cipher yang digunakan untuk proses dekripsidengan kunci yang semakin panjang pulamaka waktu yang dibutuhkan akan semakin lama.

B. SARAN

Tugas akhir ini dapat dikembangkan dengan pada *audio file* lain seperti WAV, AAC.

DAFTAR PUSTAKA

[1] **Abdurohman, Maman. 2002. Analisis Performansi Algoritma Kriptografi RC6. Bandung: Institut Teknologi Bandung.**

[2] **Ariyus, Dony. 2008. Pengantar Ilmu Kriptografi: Teori, Analisis, dan Implementasi. Yogyakarta: Penerbit Andi.**

[3] **Raissi, Rassol. 2002. *The Theory Behind Mp3*. http://www.mp3-tech.org/ programmer/docs/mp3_theory.pdf.**